

**AMENDMENTS TO THE CLAIMS**

This listing of claims will replace all prior versions of claims in the application:

**Listing of Claims:**

1. (Currently Amended) A system that facilitates a software development environment, comprising:  
~~an unmanaged application that facilitates processing a document; and~~  
a hosting component that interfaces ~~to the~~ an unmanaged application;  
a designer framework that interfaces a development environment, the hosting component and designer framework interface such that the unmanaged application functions as a designer within the development environment; and  
a host adaptor that merges at least one menu of the unmanaged application with at least one menu of the development environment, determines if the development environment has focus or if unmanaged application has focus, and enables at least one merged menu item and disabled at least one merged menu item based upon a result of the determination;  
wherein memory coupled to a processor retains the hosting component, the design framework, the host adaptor, or a combination thereof.
2. (Original) The system of claim 1, the hosting component supports an object linking and embedding (OLE) technology.
3. (Currently Amended) The system of claim 1, the unmanaged application includes at least one of a word processing application [[and]] or a spreadsheet application.
4. (Original) The system of claim 1, further comprising an integration interface to facilitate integrating a third-party unmanaged application as a designer in the development environment.

5. (Original) The system of claim 1, the development environment is an integrated development environment (IDE).
6. (Currently Amended) The system of claim 1, ~~further comprising a host adaptor that interfaces to the unmanaged application, which the host adaptor is application specific to facilitate integrating the unmanaged application functioning as a designer within into the development environment.~~
7. (Original) The system of claim 1, further comprising a document-hosting subcomponent that facilitates hosting the document that can be manipulated in the development environment.
8. (Original) The system of claim 1, the document can be manipulated using at least one of the unmanaged application and the development environment.
9. (Original) The system of claim 1, the document is one of a new document and an existing document.
10. (Cancelled)
11. (Original) The system of claim 1 facilitates hosting a plurality of different unmanaged applications.
12. (Original) A computer readable medium having stored thereon computer executable instructions for carrying out the system of claim 1.
13. (Original) A computer employing the system of claim 1.
14. (Original) The system of claim 1, further comprising a tray component that facilitates presenting non-visual data in the IDE.

15. (Currently Amended) A system that facilitates an IDE, comprising:  
~~an unmanaged application that facilitates processing a document; and~~  
a hosting component that interfaces to the unmanaged application such that the unmanaged application functions as a designer within the IDE, the hosting component further comprising,

an integration interface to facilitate integrating a third-party unmanaged application as a designer in the development environment;

a host adaptor that interfaces to the unmanaged application, ~~which the host~~ adaptor is application specific to facilitate integrating the unmanaged application into the development environment, merges at least one menu of the unmanaged application with at least one menu of the development environment, determines if the development environment has focus or if unmanaged application has focus, and enables at least one merged menu item and disabled at least one merged menu item based upon a result of the determination; and

a document-hosting subcomponent that facilitates hosting a document that can be manipulated in the development environment, the hosted document has at least two views, including a design view and a code view; and[[.]]

a designer framework that interfaces a development environment, the hosting component and designer framework interface such that the unmanaged application functions as a designer within the development environment, the hosted application operates in at least two modes including a design mode and a runtime mode to distinguish between design actions and runtime actions within the hosted application.

16. (Original) The system of claim 15, the host component further comprising a designer framework that facilitates interfacing the hosting component to the development environment.

17. (Original) The system of claim 15, wherein native functionality of the unmanaged application is available within the IDE.

18. (Original) The system of claim 15 facilitates adding a control to the document and editing properties of the control.

19. (Original) The system of claim 15 facilitates a merging of menus of the unmanaged application and the IDE.

20. (Original) The system of claim 15, the document contains at least one of embedded and linked objects.

21. (Original) The system of claim 15 facilitates the presentation of a properties window that displays properties for code-behind project items.

22. (Original) The system of claim 15, the designer operates in a design view and a code view.

23. (Original) The system of claim 15, the designer can handle the document in at least one of a binary format and an XML format.

24. (Currently Amended) A computer-readable medium having computer-executable instructions for performing a method of integrating an unmanaged application into a development environment, the method comprising:

receiving the unmanaged application; and

interfacing the unmanaged application to the development environment with a host adaptor that is specific to the unmanaged application such that the unmanaged application is accessible as a designer within the development environment, the host adaptor merges at least one menu of the unmanaged application with at least one menu of the development environment, determines if the development environment has focus or if unmanaged application has focus, and enables at least one merged menu item and disabled at least one merged menu item based upon a result of the determination.

25. (Original) The method of claim 24, further comprising hosting a document in the development environment such that the document is manipulated using native functionality of the unmanaged application and functionality of the development environment.

26. (Original) The method of claim 24, further comprising hosting a document in the development environment and exposing a code-behind file associated with the document such that contents of the file can be manipulated.

27. (Original) The method of claim 24, further comprising hosting a document in the development environment and exposing a code-behind file associated with each subdocument of the document such that contents of each file can be manipulated.

28. (Original) The method of claim 24, further comprising providing a view control that triggers in response to an event.

29. (Currently Amended) A method of integrating an unmanaged application into a development environment, comprising:

~~receiving~~ obtaining the unmanaged application;

interfacing the unmanaged application to the development environment with a host adaptor that is specific to the unmanaged application such that the unmanaged application is accessible as a designer within the development environment, the host adaptor merges at least one menu of the unmanaged application with at least one menu of the development environment, determines if the development environment has focus or if unmanaged application has focus, and enables at least one merged menu item and disabled at least one merged menu item based upon a result of the determination; and

hosting a document in the development environment such that functionality of the development environment and the unmanaged application can be used to manipulate the document.

30. (Original) The method of claim 29, further comprising manipulating the document by at least one of,

- performing a drag-and-drop operation of a control onto the document;
- viewing a data-behind file associated with the document;
- writing managed code in the data-behind file;
- operating the designer in a design mode and a runtime mode;
- presenting non-visual data; and
- drawing managed objects on the document.

31. (Original) The method of claim 29, further comprising providing a special page of preferences for the unmanaged application such that when integrated into the development environment, the unmanaged application behaves according to the preferences.

32. (Currently Amended) A system that facilitates of integrating an unmanaged application into a development environment, comprising:

- means for ~~receiving~~ obtaining the unmanaged application;

- means for interfacing the unmanaged application to the development environment with a host adaptor that is specific to the unmanaged application such that the unmanaged application is accessible as a designer within the development environment;

- means for merging at least one menu of the unmanaged application with at least one menu of the development environment, determines if the development environment has focus or if unmanaged application has focus, and enables at least one merged menu item and disabled at least one merged menu item based upon a result of the determination;

- means for hosting a document in the development environment such that functionality of the development environment and the unmanaged application can be used to manipulate the document; and

- means for manipulating the document.

33. (Original) The method of claim 32, the means for manipulating further comprising manipulating means for at least one of,

- performing a drag-and-drop operation of a control onto the document;
- viewing a data-behind file associated with the document;
- writing managed code in the data-behind file;
- operating the designer in a design mode and a runtime mode;
- presenting non-visual data; and
- drawing managed objects on the document.

34. (Original) The method of claim 32, further comprising means for exposing properties and components of the document for manipulation.